PF-MT (Particle Filter with Mode Tracker) for Tracking Contour Deformations

Namrata Vaswani, Yogesh Rathi, Anthony Yezzi, Allen Tannenbaum

***Abstract*— We consider the problem of tracking the boundary contour of a moving and deforming object from a sequence of images. If the motion of the "object" or region of interest is constrained (e.g. rigid or approximately rigid), the contour motion can be efficiently represented by a small number of parameters, e.g. the affine group. But if the "object" is arbitrarily deforming, each contour point can move independently. Contour deformation then forms an infinite (in practice, very large), dimensional space. Direct application of particle filters for large dimensional problems is impractical, due to the reduction in effective particle size as dimension increases. But in most real problems, at any given time, "most of the contour deformation" occurs in a small number of dimensions ("effective basis") while the residual deformation in the rest of the state space ("residual space") is "small". The effective basis may be fixed or time varying. Based on this assumption, we modify the particle filtering method to perform sequential importance sampling only on the effective basis dimensions, while replacing it with deterministic mode tracking in residual space (PF-MT). We develop the PF-MT idea for contour tracking. Techniques for detecting effective basis dimension change and estimating the new effective basis are presented. Tracking results on simulated and real sequences are shown and compared with past work.**

I. Introduction

A. The Problem

We would like to causally segment a moving/deforming object(s) from a sequence of images. This is formulated as a problem of tracking the boundary contour of the object, i.e. computing an "optimal" estimate of the state (contour and contour velocity) at the current time using all observations (images) until the current time. We denote the state at time t by X_t and the observation by Y_t. Any "optimal" state estimate can be computed once the posterior, $\pi_t(X_t) \triangleq p(X_t|Y_{1:t})$, is computed or approximated, e.g. MAP or MMSE. The general problem formulation is depicted in Fig. 1. The state dynamics is assumed to be Markovian. The observed image is assumed to be a noisy and possibly nonlinear function of the contour. The image likelihood given the contour ("observation likelihood") may be multimodal or heavy tailed. Since the state space model is nonlinear and multimodal, we study particle filtering(PF) [2], [3], [4] solutions to the tracking problem.

A continuous closed curve (contour)[5] is the smooth locus of points traced out by the mapping of the unit interval into $\mathbb{R}^2$. If the motion of the "object" is constrained, the contour motion can be efficiently represented by a small number of parameters, e.g. the affine group [6], [7]. But if the "object" is arbitrarily deforming, each contour point can move independently. Contour deformation then forms an infinite (in practice, very large), dimensional space. Deforming contours occur either due to changing region of partial occlusions or when the object of interest is actually deforming its shape over a time or space sequence of images. An example of the first kind is shown in Fig. 6(a), where the contour representing the left part of the car deforms as it moves under the pole. Examples of the second kind are a beating heart, moving animals or humans, or the cross-sections of different parts of a 3D object like the brain, in consecutive MRI slices, e.g. Fig. 7. Most biological images contain deforming objects/regions. Contour tracking has many applications in medical image analysis, e.g. sequential segmentation of volume images (Fig. 7); tracking heart regions [8], [9] or image guided surgery [10].

The observation likelihood is often multimodal due to background objects (clutter) which are partially occluded by the "object of interest" (for e.g. see Fig. 3) or due to an object which partially occludes the "object of interest" (for e.g. the two contour modes shown in Fig. 6(a), 6(b)) or due to low contrast imagery (e.g. see Fig. 7 or [9]). Heavy tailed and often multimodal observation likelihoods occur when the observation noise has occasional outliers (for e.g. see Fig. 4).

B. Motivation

Early work on contour tracking used the Kalman filter to track a fixed number of marker points uniformly chosen on the initial contour [11], [12], [13] or a fixed parametric representation, such as B-spline control points [14]. A Kalman filter for

N. Vaswani is with the ECE Dept. at Iowa State University, Ames, IA 50011 (email: namrata@iastate.edu). A. Yezzi, Y. Rathi and A. Tannenbaum are with the School of ECE at Georgia Tech, Atlanta, GA 30332. A part of this paper will be presented in [1]. A. Tannenbaum is also with the Department of EE, Technion, Israel. This work was partially supported by research initiation grants from Iowa State University and by grants from AFOSR, ARO, NSF, MURI, MRI-HEL, and NIH. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by NIH.

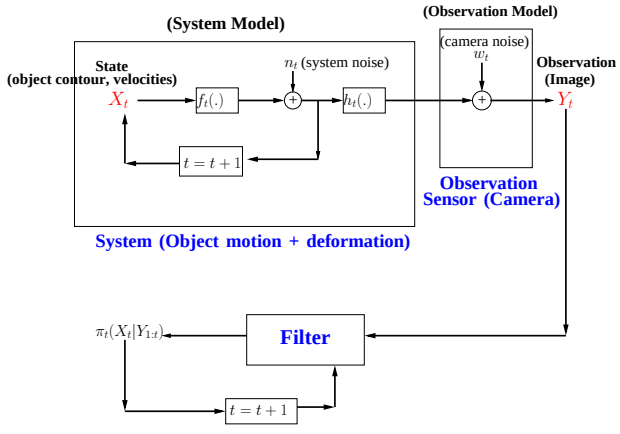


Fig. 1. Problem Formulation

a continuous contour was proposed in [15]. The Kalman filter can only handle additive and unimodal observation noise and so the observation needs to be an observed contour extracted from the image by searching in the vicinity of the predicted contour. The seminal work of [6] (Condensation) introduced particle filters (PF) [2], [3], [4] to tackle multimodal (and possibly nonlinear) observation likelihoods that occur due to clutter or occlusions. It allowed directly using the image (or the edge map) as the observation. But it only tracked on the 6-dim space of affine deformations.

Many recent works on contour tracking [16], [17], [18], [19], [20], [21], [22] use the level set representation [23] of a contour and propose different types of approximate linear observers for contour deformation and/or for global motion. The level set method [23], [24], [25] provides a way to implicitly represent and deform a continuous contour on a fixed pixel grid and thus automatically handles changes in contour length or topology. Specifically, [16] computes the current contour estimate as an approximate linear combination of the predicted contour and the observation likelihood mode nearest to it (and similarly for global motion). We call this general technique a Posterior Mode Tracker [26], since it can be understood as computing the mode, $\hat{X}_t$, of $p^*(X_t) \triangleq p(X_t|\hat{X}_{t-1}, Y_t)$ and approximating the posterior, π_t, by a Dirac delta function (δ) at $\hat{X}_t$. Setting, $\pi_{t-1}(X_{t-1}) \approx \delta(X_{t-1} - \hat{X}_{t-1})$, one can easily see that $\pi_t(X_t) \approx p^*(X_t)$ i.e. $\hat{X}_t$ is also the mode of the posterior. Thus it *implicitly assumes that the posterior is effectively unimodal* (has only one significant mode which is near $\hat{X}_{t-1}$). This assumption may not hold when there are multiple distinct or overlapping objects (background clutter/occlusions).

In [27], we combined the ideas of [6] and [16] to handle more general situations. A PF was used to track affine deformations, while an approximate linear observer was defined to estimate the non-affine deformation for each affine deformed contour particle. In doing this, the *implicit assumption is that the posterior of non-affine deformation is unimodal*. This is valid for many practical problems shown in [27] where the non-affine deformation per frame is small, e.g. a rigid object tracked by a perspective camera with frequent viewpoint changes, or approximately rigid objects, e.g. human body contour from a distance. *But in other situations, where local deformations are large, there may be more than one non-affine mode for the same affine deformation value and the same image, i.e. posterior of non-affine deformation may be multimodal.* This is demonstrated in Fig. 3 (overlapping objects separated by non-affine deformation) and in Fig. 4 (multiple observation likelihood modes due to outlier image and due to overlapping objects). Another example is the car sequence of Fig. 6, where one may want to either track the whole car or only the portion to the left of the street light (the two contour modes are separated by non-affine deformation). [27] tracked the full car by using a special occlusion handling method (which penalized deviations from a rigid car template).

C. Main Idea

To address the problems of [27], we need an importance sampling step [3] in the PF that also samples from the space of non-affine deformations. For deforming objects, each contour point can move independently and hence the contour deformation forms an infinite (in practice, very large), dimensional space. PF on such a large dimensional space is impractical due to the reduction in effective particle size [3] as dimension increases. But in most real problems, *at any given time, "most of the contour deformation" occurs in a smaller number of dimensions ("effective basis") while the deformation in the rest of the state space (space of "residual deformations") is "small". The effective basis may be fixed or time varying.* This is the "large dimensional state spaces (or LDSS)" property, introduced in [26], [28], applied to deforming contours. It is demonstrated in Fig. 2 where the contour dimension is $M = 178$, but "most of the contour deformation" is described by only $K = 12$ basis points. In other words, the deformation "signal" is approximately bandlimited (spatially), with the approximate cut-off frequency being much smaller than the maximum measurable frequency, 0.5Hz (cycles/pixel).

Using the LDSS property, we proposed to modify the PF method to perform sequential importance sampling [3] only on the effective basis dimensions, while replacing it with deterministic mode tracking (MT) in residual space [26], [28]. In this work, we develop the PF-MT idea for contour tracking using global translation and deformation velocity at subsampled contour locations interpolated using a B-spline basis as the effective basis. Detecting change in effective basis dimension (when contour length or deformation frequency changes) and estimating the new effective basis is also discussed. In practice, explicitly tracking local deformation (even with deformation velocity tracked at only $K = 6$ locations around the contour) is extremely beneficial as can be seen from the last rows of Figs. 3 and 4.

D. Relation to Existing Work

We stress the difference from Condensation [6] which uses B-spline control points to approximate the contour itself and hence requires many more sample points for accurate representation. This is because the maximum spatial frequency of deformation cannot be larger (is usually much smaller) than that of the two contours[1] from which it is computed.

The work of [27] can be understood as an Affine PF-MT (uses space of affine deformations as the effective basis and all non-affine deformation is treated as residual deformation). Condensation [6], [7] can be explained as Affine PF-MT with zero residual deformation. The approximate linear observer (or posterior mode tracker [26]) of [16], and also other related methods [18], [19], [21], [17], [22], may be explained as PF-MT with a zero dimensional effective basis. The algorithm of [9] can be understood as PF-MT with the difference that it retains only the MAP particle at the end of each time step and it uses a PCA effective basis [29], [30]. Our effective basis is similar to that of [31] which proposed an annealing based technique for segmentation.

Our proposed algorithms may also be used with other interpolation functions (other than B-spline) or other types of

[1]The radius of the osculating circle [5] as a function of arclength is treated as the contour "signal". Deformation (along the contour normal) is the difference of the two consecutive contour "signals". When performing a linear operation, new frequencies cannot be introduced.

effective basis, e.g. the PCA basis [29], [30], [9], [22]. Also, in this work we use a level set representation of the contour [24], [25]. Our approach is also directly applicable if this is replaced by an explicit contour parametrization [11], which is frequently changed as contour length changes.The main idea of PF-MT can also be applied to other problems such as tracking optical flow or tracking illumination change of moving objects.

Another approach to track deformations uses exemplars, e.g [32], [20]. But this can handle only a few types of deformations (depends on number of available exemplars). There is also a large amount of work on tracking [33], [34], [35], [36], [37] that does not use the "optimal filter" formulation (i.e. does not compute "optimal" posterior estimates).

Another PF method that also improves effective particle size by reducing PF dimension is Rao-Blackwellization [38], [39]. But it requires that a part of the state have a linear Gaussian state space model, which is not true for our problem. PFs with time-varying dimension have also been used in [40], [41].

The paper is organized as follows. We give the form of the state space model in Section II. The PF-MT algorithm for contour tracking and PF-MT-TV for dealing with time-varying effective basis is explained in Section III. Methods for learning the effective basis dimension and the system model parameters are discussed in Section IV. Experimental results on simulated and real sequences are given in Section V. Conclusions and open issues are discussed in Section VI.

II. State Space Model

The observation at time t (image and edge map at t) is denoted by Y_t and the state at t (contour, contour velocity) is denoted by X_t. A block diagram is shown in Fig. 1. The contour at t can be represented as $C_t = C_t(p) = [C_t^x(p), C_t^y(p)]$, $p \in [0,1]$. The parametrization is not unique, i.e. all re-parameterizations of the parameter p of the form $\tilde{p} = f(p)$, where $f : [0,1] \rightarrow [0,1]$ is continuous and strictly monotonic, yield the same contour [5]. The outward normal to contour C_t at p is denoted by $\overrightarrow{\mathbf{N}}(C_t(p))$ or by $\overrightarrow{\mathbf{N}}_t(p)$. Denote the space of contours [42] by $\mathcal{S}$. Then the tangent space to $\mathcal{S}$ at C_t will be [42] the space of all non-tangential velocities (velocities along the normal to C_t at each point), since tangential velocity only re-parameterizes the contour [5]. We use v_t to denote the vector of normal velocities.

We use the notation $\mathcal{N}(x; \mu, \Sigma) \triangleq \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-x^T \Sigma^{-1} x}$.

Note that the contour is a geometric entity [5], i.e. it is a function of its own arclength. Any deformation of the contour deforms the arclength. In implementation using a level set method [24], [25], this results in change in contour dimension, M_t, and also change in distance between neighboring points, i.e. $p_j, j = 1, 2, ..M_t$ are not uniformly spaced.

A. System Model

The state at any time t, consists of the contour, its normal deformation velocity, and the global translational velocity. Because of the LDSS property (described in Section I-C), "most of the contour deformation" occurs in a smaller number of dimensions, K, which form the "effective basis". Thus, we split v_t as $v_t(p) = B_s(p)v_{t,s} + B_r(p)v_{t,r} + \overrightarrow{\mathbf{N}}_t(p)^T \rho_t$ where B_s denotes the effective basis directions for contour deformation (with translation removed) while B_r denotes the basis spanning the residual space. $v_{t,s}$, $v_{t,r}$ denote the corresponding coefficients. $\rho_t \in \mathbb{R}^2$ denotes the global x-y translation vector. We use velocity at K subsampled locations interpolated onto the entire contour using B-spline interpolation functions as the effective basis. This is explained in Section II-C. We assume that ρ_t, $v_{t,s}$ follow a first order autoregressive (AR) model, while $v_{t,r}$ is assumed temporally independent. Thus, the system dynamics of $X_t = [C_t, v_{t,s}, v_{t,r}, \rho_t]$ is:

$$\begin{aligned}
\partial C_t(p) &= [B_s(p)dv_{t,s} + B_r(p)dv_{t,r} + \overrightarrow{\mathbf{N}}_t(p)^T d\rho_t]\ \overrightarrow{\mathbf{N}}_t(p) \\
dv_{t,s} &= -A'_s v_{t,s} dt + dW_{t,s}, \quad dW_{t,s} \sim \mathcal{N}(0, \Sigma'_s) \\
dv_{t,r} &= dW_{t,r}, \quad dW_{t,r} \sim \mathcal{N}(0, \Sigma'_r) \\
d\rho_t &= -A'_\rho \rho_t dt + dW_{t,\rho}, \quad dW_{t,\rho} \sim \mathcal{N}(0, \Sigma'_\rho) \qquad (1)
\end{aligned}$$

where $\partial C_t(p)$ denotes partial differential of C_t w.r.t. time and $W_{t,s}, W_{t,r}, W_{t,\rho}$ are Brownian motions [43], with dimensions K, $M_t - K$ and 2 respectively. $dW_{t,s}$ (similarly $dW_{t,r}, dW_{t,\rho}$) denotes the change in $W_{t,s}$ in time dt. Also, $0 < A'_s < I$ and $0 < A'_\rho < I$. Assume that the observations arrive every τ time instants. The above differential equations can be discretized as follows. We remove "(p)" for clarity.

$$C_n = C_{n-1} + [B_s v_{n,s} + B_r v_{n,r} + \overrightarrow{\mathbf{N}}_{n-1}^T \rho_n]\ \overrightarrow{\mathbf{N}}_{n-1} \quad (2)$$

$$v_{n,s} = [I - A'_s\tau]v_{n-1,s} + \nu_{n,s}, \quad \nu_{n,s} \sim \mathcal{N}(0, \Sigma'_s\tau^2) \quad (3)$$

$$v_{n,r} = \nu_{n,r}, \quad \nu_{n,r} \sim \mathcal{N}(0, \Sigma'_r\tau) \quad (4)$$

$$\rho_n = [I - A'_\rho\tau]\rho_{n-1} + \nu_{n,\rho}, \quad \nu_{n,\rho} \sim \mathcal{N}(0, \Sigma'_\rho\tau^2) \quad (5)$$

The discretization given in (2) assumes that

Assumption 1: The observation interval τ is small enough (compared to $B_s v_{n,s} + B_r v_{n,r} + \overrightarrow{\mathbf{N}}_{n-1}^T \rho_n$) so that $\overrightarrow{\mathbf{N}}_{n-1}$ is also approximately normal to C_n.

Defining $\Sigma_s \triangleq \Sigma'_s\tau^2$, $\Sigma_\rho \triangleq \Sigma'_\rho\tau^2$, $\Sigma_r \triangleq \Sigma'_r\tau$, $A_s \triangleq [I - A'_s\tau]$, $A_\rho \triangleq [I - A'_\rho\tau]$, and $\tilde{C}_n \triangleq C_{n-1} + [B_s v_{n,s} + \overrightarrow{\mathbf{N}}_{n-1}^T \rho_n]\overrightarrow{\mathbf{N}}_{n-1}$, and again removing "$(p)$", we can rewrite the system model:

$$C_n = \tilde{C}_n + B_r v_{n,r} \overrightarrow{\mathbf{N}}(\tilde{C}_n) \quad (6)$$

$$\tilde{C}_n = C_{n-1} + [B_s v_{n,s} + \overrightarrow{\mathbf{N}}_{n-1}^T \rho_n]\ \overrightarrow{\mathbf{N}}_{n-1}, \quad B_s \triangleq B_s(C_{n-1}) \quad (7)$$

$$v_{n,s} = A_s v_{n-1,s} + \nu_{n,s}, \quad \nu_{n,s} \sim \mathcal{N}(0, \Sigma_s) \quad (8)$$

$$v_{n,r} = \nu_{n,r}, \quad \nu_{n,r} \sim \mathcal{N}(0, \Sigma_r) \quad (9)$$

$$\rho_n = A_\rho \rho_{n-1} + \nu_{n,\rho}, \quad \nu_{n,\rho} \sim \mathcal{N}(0, \Sigma_\rho) \quad (10)$$

where $B_s \triangleq B_s(C_{n-1})$ is defined by (16) or by (17) given in Appendix A. Note, $v_{n,r}$ is actually not part of the state vector (since no element of the next state, X_{n+1}, depends on $v_{n,r}$).

B. Level Set Representation

We use the level set method [24], [25] since it automatically handles contour length changes. C_t is represented implicitly as the zero level set of a higher dimensional function, denoted $\phi_t(x)$, i.e. C_t is the collection of all points $\{x \in \mathbb{R}^2 : \phi_t(x) = 0\}$ [24], [25] (x denotes the x-y coordinates). The direction of the gradient of ϕ_t, $\nabla\phi_t(x)$, is along the normal, $\overrightarrow{\mathbf{N}}_t$. The most common choice of $\phi_t(x)$ is the "signed distance function" [24], [25], i.e. the magnitude of $\phi_t(x)$ is the minimum of the distance of x from any contour point

and its sign is taken to be negative inside the contour and positive outside the contour. Level set evolution corresponding to contour evolution of the form $\frac{\partial C_t}{\partial t} = v\overrightarrow{\mathbf{N}}$ is given by[24], [25] $\frac{\partial \phi_t(x)}{\partial t} = v_{extend}(x)||\nabla\phi_t||$ where v_{extend} is the normal extension [24], [25] of v onto non-zero level sets.

C. Geometric and Parametric Effective Basis

Contour motion using the level set method is naturally implemented using a B-spline basis that parameterizes velocity of a contour point based on its location on the x-y plane *(geometric effective basis)*. There are many possible ways to define a geometric basis, e.g. see [31], [44]. For example, one dimensional parameterizations can be obtained by using the turning angle (angle made by the tangent with the x axis) or the radial angle (angular coordinate of the contour point w.r.t. the centroid of the contour's inside region, $[\mu_n^y, \mu_n^y]$) as the parameter. In our implementations, we use the radial angle as the parameter. See Appendix A for the precise details and equations. A velocity sample $v_{n,s,j}, j = 1, 2, ..K$ is assigned to each angular region. For e.g., a four dimensional basis is obtained by allocating one velocity sample to one quadrant of the x-y plane and smoothing across quadrant boundaries using B-spline interpolation. A geometric basis automatically handles changes in contour topology.

But *it cannot be used if one would like to independently deform two or more points of a contour that have the same radial angle, but are far if one moves along the contour arclength.* Such applications can be handled by a *parametric effective basis* which parameterizes velocity of a contour point based on its arclength location w.r.t. a fixed initial point. The basis points split the contour arclength into K regions and a velocity sample, $v_{n,s,j}, j = 1, 2, ..K$, is assigned to each region. See Appendix A for the precise details and equations. The level set implementation issues, explained in Section III-A, will further clarify things. The parametric basis is useful for applications where change in topology is not allowed. In practical implementations, we can detect topology change of contour particles and assign a zero likelihood to particles for which topology change occurs.

D. Observation Model

The observation at time n, Y_n, is the image at n and the edge map derived from it. We assume that Y_n, depends only on C_n (and not on the velocity), i.e. the observation likelihood, $p(Y_n|X_n) = p(Y_n|C_n)$. Many observation models have been proposed in literature for segmentation and tracking - these can be classified as "region based", e.g. [45], [27], [9], or "edge based", e.g. [6] or "motion based", e.g. [13].

Using a good observation model is a critical issue, but we have not addressed it here. In this paper, we use a product of the simple region-based observation likelihood of [27] which was motivated by the Chan and Vese model [45] and the edge-based observation likelihood proposed in Condensation [6]. This combines the advantages of a region based approach (robustness to blurred edges and ability to select the object of interest) with those of an edge based approach (ability to deal with intensity variations across the sequence and with errors in learning the foreground or background object intensities). The implicit assumptions in using this model are discussed in Appendix B. The combined model is multimodal with a strong mode at the object of interest (due to high region and edge likelihood) and a weaker mode at any "object" (due to high edge likelihood only).

III. PF-MT (Particle Filter with Mode Tracker)

We first explain a generic particle filtering (PF) algorithm [2], [4], [3]. A PF outputs at each time n, a cloud of N particles, $\{X_n^i\}$ with weights $\{w_n^i\}$ whose empirical measure $\pi_n^N(X_n) \triangleq \sum_{i=1}^N w_n^i \delta(X_n - X_n^i)$ closely approximates the true posterior, $\pi_n(X_n) \triangleq p(X_n|Y_{1:n})$. Here $\delta(X-a)$ denotes the Dirac delta function at a. It starts with sampling N times from π_0 at $n = 0$ to approximate it by $\pi_0^N(X_0)$. For each $n > 0$, it approximates the Bayes recursion for going from π_{n-1}^N to π_n^N using importance sampling. This consists of:

1) *Importance Sampling (IS):* Sample $X_n^i \sim q(X_n^i)$, for $i = 1, 2...N$. The importance sampling density, q, can depend on X_{n-1}^i and Y_n.
2) *Weighting:* Compute the weights: $w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^N \tilde{w}_n^{(j)}}$, where $\tilde{w}_n^i = w_{n-1}^i \frac{p(Y_n|X_n^i)p(X_n^i|X_{n-1}^i)}{q(X_n^i)}$.
3) *Resampling:* Replicate particles in proportion to their weights and reset w_n^i [3].

Since the effective particle size decreases with increasing system noise dimension, direct application of PF becomes impractical for large dimensional problems. *We propose to replace the PF by the following: importance sample only on the effective basis dimensions, and replace the importance sampling step by a deterministic Mode Tracking (MT) step in the residual space [26].* This idea, which we call PF-MT, assumes that the effective basis dimension K is large enough to ensure that Assumptions 2 and 3, given below, hold [26].

Assumption 2: The total residual deformation variance, $\Delta_{tot} = trace(\Sigma_r)$ is small enough so that the posterior in residual space, $p^{**,i}$ (defined below), is unimodal.

$$\begin{aligned} p^{**,i}(v_{n,r}) &\triangleq p(v_{n,r}|v_{n,s}^i, \rho_n^i, X_{n-1}^i, Y_n) \\ &\propto p(Y_n|\tilde{C}_n^i + B_r v_{n,r}\overrightarrow{\mathbf{N}})\ \mathcal{N}(v_{n,r}; 0, \Sigma_r) \\ &\triangleq \exp[-L^i(v_{n,r})] \end{aligned} \tag{11}$$

When Assumption 2 holds, one can use the following importance sampling strategy [26]: sample $v_{n,s}^i$, ρ_n^i from their state transition priors, $\mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$, $\mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ respectively; compute $\tilde{C}_n^i$ using (7); and sample $v_{n,r}^i$ from a Gaussian approximation [46], denoted $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$, to $p^{**,i}$ about its mode, m_n^i. Finally, compute C_n^i using (6). Here

$$m_n^i \triangleq \arg\min_{v_{n,r}} L^i(v_{n,r}), \quad \Sigma_{IS}^i \triangleq (\nabla^2 L^i(v_{n,r}))^{-1} \tag{12}$$

Now, by conditional variance identity [47], $\mathbb{E}_{Y_n}[\Sigma_{IS}^i] \approx \mathbb{E}_{Y_n}[Variance(p^{**,i})] \leq \Sigma_r$. Thus if the following assumption holds, we can replace importance sampling from $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$ by [26] deterministically setting $v_{n,r}^i = m_n^i$. We call this the Mode Tracking (MT) approximation.

Assumption 3: The total residual deformation variance, $\Delta_{tot} = trace(\Sigma_r)$ is small enough, i.e. $\Delta_{tot} < \Delta_{tot,bnd}$, so

Algorithm 1 PF-MT (Exact). Going from π_{n-1}^N to $\pi_n^N(X_n) = \sum_{i=1}^N w_n^{(i)} \delta(X_n - X_n^i)$, $X_n^i = [C_n^i, \rho_n^i, v_{n,s}^i, v_{n,r}^i]$

1) *Importance Sample (IS) on effective basis:* $\forall i$, sample $v_{n,s}^i \sim \mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$, sample $\rho_n^i \sim \mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ and compute $\tilde{C}_n^i$ using (7).
2) *Mode Tracking (MT) in residual space:* $\forall i$,
   a) Compute m_n^i, Σ_{IS}^i using (12).
   b) Set $v_{n,r}^i = m_n^i$. **[instead of Importance Sampling from $\mathcal{N}(m_n^i,\ \Sigma_{IS}^i)$]**
   c) Compute C_n^i using (6).
3) *Weight & Resample:* Compute w_n^i using (14) and resample [3].

Algorithm 2 PF-MT (Approximate). Going from π_{n-1}^N to $\pi_n^N(X_n) = \sum_{i=1}^N w_n^{(i)} \delta(X_n - X_n^i)$, $X_n^i = [C_n^i, \rho_n^i, v_{n,s}^i, v_{n,r}^i]$

1) *Importance Sample (IS) on effective basis:* $\forall i$, sample $v_{n,s}^i \sim \mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$, sample $\rho_n^i \sim \mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ and compute $\tilde{C}_n^i$ using (7).
2) *Mode Tracking (MT) in residual space:* $\forall i$, compute C_n^i by starting with $\tilde{C}_n^i$ and running k iterations of Gradient Descent to minimize $E(C_n) \triangleq -\log[p(Y_n|C_n)]$. In experiments, $k = 1$ or $k = 2$ suffice.
3) *Weight & Resample:* Compute w_n^i using (15) and resample [3].

that with high probability, there is little error in replacing a random sample from $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$ by m_n^i.

Based on the above ideas, we develop the PF-MT algorithm for contour tracking [1]. We explain each step below and the approximations used to simplify the implementation. The exact stepwise algorithm is given in Algorithm 1 and the approximations are summarized in Algorithm 2.

A. Importance Sampling on Effective Basis Dimensions

This involves sampling $v_{n,s}^i$ and ρ_n^i from their state transition priors, $\mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$ and $\mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ respectively and computing $\tilde{C}_n^i$ using (7), $\forall i = 1, 2, ..N$. We implement (7) using the level set method. The level set evolution corresponding to contour evolution given by (7), is: $\tilde{\phi}_n^i(x) = \phi_{n-1}^i(x) + v_{extend}(x)||\nabla_x \phi_{n-1}^i(x)||$ where v_{extend} is the normal extension [24], [25] of $B_s(p) v_{n,s}^i + \overrightarrow{\mathbf{N}}_{n-1}^T(p) \rho_n^i$ onto non-zero level sets. This implementation assumes that Assumption 1 holds. If it does not hold (either τ large or motion fast), then (7) will have to be implemented using multiple iterations. If the narrowband level set method [24], [25] is used, multiple iterations may be required to implement (7), depending on the velocity magnitude and the narrowband width. Since we do not evolve a PDE to convergence, CFL condition [24], [25], [5] need not be satisfied.

B_s and its extension onto all level sets need to be computed at each step. This can be done without computing the zero level set (contour) for the geometric basis[2]. For the parametric basis, at each iteration, (i) the contour (zero level set) needs to be computed; it needs to always be traversed in the same order (say clockwise); and starting point correspondence needs to be maintained; (ii) the basis points need to be moved along with the contour using (20) given in Appendix A and (iii) the B-spline interpolation functions need to be recomputed using the current arclength distance between the basis points. If basis points come too close or go too far, the basis needs to change.

[2]assuming the requirement of normal extension velocities [24], [25] is relaxed (in numerical implementation, this will only result in more frequent re-initialization of level set functions since embedded curves may not remain embedded [5]). If turning angle is used as the parameter, then one can directly get normal extension velocities at each point without computing the contour.

B. Mode Tracking on Residual Space

This involves computing the mode m_n^i defined in (12), setting $v_{n,r}^i = m_n^i$; and computing C_n^i using (6), $\forall i = 1, 2..N$. Computing m_n^i and C_n^i requires being able to compute B_r. But B_r is the solution of $B_r B_r^T = I - B_s(B_s^T B_s)^{-1} B_s^T$. Since B_s depends on C_{n-1}^i, it will need to be computed at each n and for all i, which is very expensive. By using some approximations, we avoid having to compute B_r.

Define $E(C_n) \triangleq -\log[p(Y_n|C_n)]$. We have shown [26] that if Assumption 2 holds, m_n^i can be computed by starting with $v_{n,r} = 0$ as initial guess and running k iterations (for some k) of gradient descent to minimize $E(\tilde{C}_n^i + B_r v_{n,r})$ w.r.t. $v_{n,r}$. When $trace(\Sigma_r)$ is small (Assumption 3 holds), maximum value of k will also be small and a heuristic choice of k suffices. If we also allow change along B_s, the k gradient descent iterations to minimize E as a function of $v_{n,r}$ can be replaced by k gradient descent iterations to minimize E as a function of C_n (skips the need to compute B_r). This approximation assumes that:

Assumption 4: We replace (6), (9) by the following model:

$$C_n = \tilde{C}_n + v_{n,r} \overrightarrow{\mathbf{N}}(\tilde{C}_n),\ v_{n,r} \sim \mathcal{N}(0, \Sigma_r),\ \Sigma_r = \Delta I \quad (13)$$

Gradient descent is implemented using the standard level set method[24], [25]. We start with $\phi^{(1)} = \tilde{\phi}_n^i$ (level set function corresponding to $\tilde{C}_n^i$) as initial guess and run k iterations of gradient descent to minimize E, i.e. run k iterations of the type $\phi^{(r+1)} = \phi^{(r)} + v_{extend}||\nabla_x \phi^{(r)}||$ where v_{extend} is normal extension [24], [25] of $(\nabla_C E)$ onto non-zero level sets. The output after k iterations is ϕ_n^i and its zero level set is C_n^i.

C. Weighting and Resampling

This involves computing the weights, w_n^i, $\forall i = 1, 2..N$, and resampling [3]. w_n^i is computed as:

$$w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^N \tilde{w}_n^j},\ \tilde{w}_n^i \triangleq w_{n-1}^i \frac{p(Y_n|C_n^i)\ \mathcal{N}(v_{n,r}^i; 0, \Sigma_r)}{\mathcal{N}(v_{n,r}^i; m_n^i, \Sigma_{IS}^i)}. \quad (14)$$

Using Assumption 4, the above only requires knowing $||v_{n,r}^i||^2$. Since in the mode tracking step we minimize directly

Algorithm 3 PF-MT-TV: PF-MT for a Time Varying Effective Basis. Going from π_{n-1}^N to $\pi_n^N(X_n)$, $X_n^i = [C_n^i, \rho_n^i, v_{n,s}^i, v_{n,r}^i]$

1) Run either Algorithm 1 or Algorithm 2.
2) *Detect Basis Change:* Detect if basis change required. If yes, go to step 3, else $n \longleftarrow n+1$ and go to step 1.
3) *Change Basis:*
   a) Compute $K_{new} = L/\alpha_s$ where L is the length of the most likely contour particle.
   b) $\forall i$, reallocate the knots uniformly and evaluate the new basis $B_{K_{new}}(C_n^{(i)}) \triangleq B_{K_{new},i}$.
   c) $\forall i$, project $v_{n,s}^i$ into the new basis: $v_{n,s}^i \longleftarrow (B_{K_{new},i}^T B_{K_{new},i})^{-1} B_{K_{new},i}^T B_i v_{n,s}^i$.
   d) $\forall i$, set $B_i \longleftarrow B_{K_{new},i}$.
   e) $n \longleftarrow n+1$ and go to step 1.

over C_n^i, we never compute $v_{n,r}^i$. We can replace $||v_{n,r}^i||$ by any easily computable distance, d, between C_n^i and $\tilde{C}_n^i$ (or between ϕ_n^i and $\tilde{\phi}_n^i$) without much error in practice. In our experiments, we use the set symmetric distance [48]. Also, since $\mathbb{E}_{Y_n}[\Sigma_{IS}^i] \leq \Sigma_r$, when Σ_r is small, one can replace Σ_{IS}^i by Σ_r. This makes the denominator a constant (can be removed). Thus, the weights can be computed as:

$$w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^N \tilde{w}_n^j}, \ \tilde{w}_n^i = w_{n-1}^i p(Y_n|C_n^i) \ e^{-\frac{d^2(C_n^i, \tilde{C}_n^i)}{\Delta}} \quad (15)$$

The exact PF-MT method is summarized in Algorithm 1 and the approximations explained above are given in Algorithm 2.

D. Time-Varying Effective Basis: PF-MT-TV

The effective basis dimension needs to be large enough so that the mode tracking approximation in residual space can be justified at each n, i.e. we need to satisfy Assumptions 2 and 3 at each n. If the contour length changes significantly, the number of basis points will need to be changed to satisfy these assumptions. Assume in this section, that we know the maximum allowable value of the distance between consecutive basis points, α_s, to ensure that $\Delta_{tot} < \Delta_{tot,bnd}$. We explain how to choose α_s in Section IV-A.

As the contour deforms, both its total length and arclength distance between consecutive basis points changes. For the parametric basis, there is a need to change effective basis if this distance becomes significantly smaller (starting to estimate noise)[3] or significantly larger (residual deformation too large) than α_s. This is done as follows. Choose a $\alpha_{s,min} < \alpha_s$ and $\alpha_{s,max} > \alpha_s$. We declare a need to change dimension, whenever the following occurs for "most" (more than 50%) of the contour particles: the arclength distance between any two consecutive basis point locations exceeds $\alpha_{s,max}$ or goes below $\alpha_{s,min}$. We evaluate the new effective basis as follows: Compute $K = \lceil L/\alpha_s \rceil$ for the most likely contour particle. Uniformly allocate the K basis points on the arclength of all contour particles and compute the new effective basis functions. Note that in certain situations, K may remain the same, but the basis points may just get placed uniform distance apart on arclength. When the effective basis changes, the old velocity coefficients' vector, $v_{n,s}^{(i)}$, also needs to be projected into the new effective basis. In this paper, we show results for varying K for only the parametric basis. Changing α_s and K for the geometric basis is briefly discussed in Section IV-B. The details are ongoing work.

The complete PF-MT algorithm with a Time-Varying effective basis (PF-MT-TV) is summarized in Algorithm 3. If the following assumptions hold, one *should be able to* show convergence of Algorithm 3 with N by extending standard results such as [49]. Weaker assumptions are discussed in [28].

Assumption 5: Assume that there is

1) No error in estimating the new effective basis dimension.
2) No delay in detecting the need for effective basis change.
3) No error in projecting $v_{n,s}^{(i)}$ into the new basis.
4) PF-MT (Algorithm 1) converges as $\Delta_{tot,bnd} \to 0$

IV. Discussion: Learning the System Model

In the experiments in this paper, we set α_s and the system model parameters $A_s, \ \Sigma_s, \ A_\rho, \ \Sigma_\rho, \ \Sigma_r$, to be equal to the ones used for simulating the sequences. For the real sequences, the choices were made by experimenting with different values. We explain here how to estimate α_s (and hence K) and the system model parameters using training sequences.

A. Learning α_s and K

We explain how to use spatial frequency of contour deformation to estimate α_s (and hence K), s.t. the total residual deformation, $\Delta_{tot} = trace(\Sigma_r) < \Delta_{tot,bnd}$ for a given value of $\Delta_{tot,bnd}$. Evaluating $\Delta_{tot,bnd}$ to satisfy Assumptions 2 and 3 is discussed in [26]. It cannot be computed in practice for our problem (since B_r cannot be computed).

We treat deformation as a function of contour arclength, or of the radial angle, and compute its Fourier transform [50]. If the deformation is temporally identically distributed (either i.i.d. or stationary), one can use a set of training vectors, $\{v_n\}_{n=1}^T$, to estimate the power spectral density (PSD) as the average of the squared Fourier transform magnitude of each v_n. We compute f_{min} as the minimum frequency such that the total power (sum of PSD values) outside f_{min} is smaller than $\Delta_{tot,bnd}$. Using Nyquist's criterion [43, pg 378], the maximum allowable distance between basis points (assumes perfect interpolation) to estimate deformation at all frequencies, $f < f_{min}$ is $\alpha_s = 1/(2f_{min})$. For total contour arclength L (or total angular "length" $L = 2\pi$), this corresponds to number of basis points, $K = \lceil L/\alpha_s \rceil = \lceil L.2f_{min} \rceil$. A PSD computed using one v_n (treated as a function of arclength) is shown in Fig. 2. f_{min} is chosen as the minimum frequency s.t. total

[3]Assumptions 2 and 3 require $\Delta_{tot} < \Delta_{tot,bnd}$ which only translates to an upper bound on the distance between consecutive basis points, α_s. But, in practice, if the distance between basis points becomes too small, the PF starts estimating noise (demonstrated in Fig. 5) because the velocities at the different basis points are assumed to be uncorrelated as explained in Section IV-D. Thus, distance becoming too small also needs to be detected and corrected.

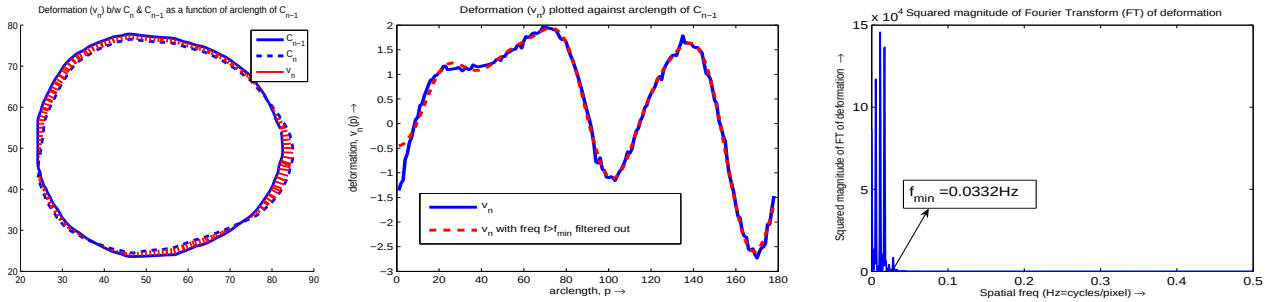


Fig. 2. Choosing K using spatial frequency. Column 1: C_{n-1}, C_n and v_n plotted on C_{n-1}. Column 2: v_n (solid blue) plotted as a function of arclength of C_{n-1} and its reconstruction (dashed red) by keeping frequencies $f \leq f_{min}$. Column 3: squared Fourier Transform of v_n. For $\Delta_{tot,bnd} = 0.05\%$ of total deformation, $f_{min} = 0.0332Hz$. $L = M = 178$ and so $K = \lceil L \cdot 2f_{min} \rceil = 12$.

power outside it is smaller than $\Delta_{tot,bnd} = 0.05\%$ of total deformation. The actual v_n and v_n with frequencies $f > f_{min}$ filtered out look almost alike visually, even though $K = 12$ while $M = L = 178$.

Note that $v_n(p)$ is computable only at points where C_{n-1} intersects the pixel grid (result of using the level set method [24], [25]), i.e. $p = p_1, p_2, ..p_{M_{n-1}}$ are non-uniformly spaced. We need to resample all v_n vectors used for computing PSD to a fixed number of uniformly spaced points before computing the Discrete Fourier Transform. For the geometric basis, the points can be spaced L/M angular distance apart with $L = 2\pi$. M can be anything (depending on the desired frequency resolution), for e.g. $M = mean(M_n)$. For the parametric basis, $L = mean(L_n)$ where L_n is the length of the contour[4] at time n. For Fig. 2, we used $M = L = L_n$.

B. Temporally Nonstationary Deformation

In the most general case, the frequency response of contour deformation may change over time. This will require also re-estimating α_s by using a few previously tracked v_n vectors to compute the new PSD and the new f_{min}.

For a geometric basis, deformation is a function of radial angle and so the total angular "length" $L = 2\pi$ remains fixed. Thus the only time K changes is when $\alpha_s = 1/(2f_{min})$ changes. Changes in contour length can result in changes in frequency response of the deformation as a function of the radial angle. This needs to be detected and α_s, K re-estimated.

C. Spatially Nonstationary Deformation

In most of the examples that we track, one region of the contour deforms more than the others, i.e. v_n is spatially nonstationary. We treat the $[-\alpha_s/2, \alpha_s/2]$ interval about a basis point as a spatially stationary region; learn the system model parameters $a_{s,j}$, $\sigma^2_{s,j}$ for each such region, j, and set $A_s = diag(a_{s,j})$ and $\Sigma_s = diag(\sigma^2_{s,j})$. Note, this implicitly assumes that the velocities at the different basis points are uncorrelated. At an effective basis change time, $a_{s,j}, \sigma^2_{s,j}$ are learnt using a few previously tracked v_n vectors.

In general, it will be more efficient to also estimate $\alpha_{s,j}$ and the number of basis points K_j for each region separately, for e.g. regions that deform negligibly may be assigned $K_j = 0$.

[4]amounts to shrinking/stretching the spatial axis on which v_n is defined from length L_n to L, followed by uniformly sampling L/M arclength distance apart. This changes the frequency content of v_n and hence is valid only for a set of approximately equal length contours (small shrinking/stretching).

D. Learning A_s, Σ_s, A_ρ, Σ_ρ, Σ_r

The basis velocities are assumed to be spatially uncorrelated[5], i.e. A_s, Σ_s are diagonal. If the deformation were exactly bandlimited; the distance between basis points were the always the Nyquist interval and if B_s were a perfect interpolator (sinc function), the basis velocities would be truly uncorrelated. All of these hold only approximately in our problem.

We describe the parameter estimation algorithm below.

1) Estimate v_n, ρ_n, $v_{n,s}, v_{n,r}$, $\forall n$: Given a sequence of approximately equal length contours $C_1, C_2, ..C_T$ and the effective basis dimension K, we first estimate ρ_n, v_n, $v_{n,s}, v_{n,r}$, $\forall n$ as explained below.

Estimate ρ_n: For a pair of contours, C_{n-1}, C_n, ρ_n is the difference between the centroid of the region inside C_{n-1} and that of the region inside C_n. Let the centered contour (contour with centroid subtracted) be denoted by C^c_n.

Estimate v_n: For each point, p, on $C^c_{n-1}(p)$, set $C^c_{n,perp}(p)$ as the point on C^c_n at which $\overrightarrow{\mathbf{N}}_{n-1}(p)$ (normal to C^c_{n-1}) intersects it. Doing this for each p on C^c_{n-1} gives $C^c_{n,perp}$ which has the same dimension, M_{n-1} and should ideally be perpendicular to $\overrightarrow{\mathbf{N}}_{n-1}$. This assumes Assumption 1. Due to numerical error, $C^c_{n,perp}(p) - C^c_{n-1}(p)$ may not be exactly along $\overrightarrow{\mathbf{N}}_{n-1}(p)$. Thus, we compute $v_n(p) = \overrightarrow{\mathbf{N}}_{n-1}(p)^T [C^c_{n,perp}(p) - C^c_{n-1}(p)]$.

Estimate $v_{n,s}$: Compute $B_s(C_{n-1})(p)$ using (16) (geometric basis) or (17) (parametric basis). Then compute an MMSE estimate of $v_{n,s}$ from v_n as $v_{n,s} = (B_s^T B_s)^{-1} B_s^T v_n$ where B_s is a matrix whose j^{th} row is $B_s(C_{n-1})(p_j)$.

Estimate $v_{n,r}$: Compute $v_{n,r}(p) = v_n(p) - B_s(p) v_{n,s}$. This assumes Assumption 4.

2) Learning A_s, Σ_s, A_ρ, Σ_ρ, Σ_r: Next, we need to learn the parameters.

Learn A_s, Σ_s using $\{v_{n,s}\}$: Estimate[43] $a_{s,j} = R_1/R_0$ where $R_1 \triangleq \frac{1}{T-1} \sum_{n=2}^{T} v_{n,s,j} v_{n-1,s,j}$ and $R_0 \triangleq \frac{1}{T} \sum_{n=1}^{T} v^2_{n,s,j}$. Compute $\sigma^2_{s,j} = \frac{1}{T-1} \sum_{n=2}^{T} [v_{n,s,j} - a_{s,j} v_{n-1,s,j}]^2$. Set $A_s = diag(a_{s,j})$ and $\Sigma_s = diag(\sigma^2_{s,j})$.

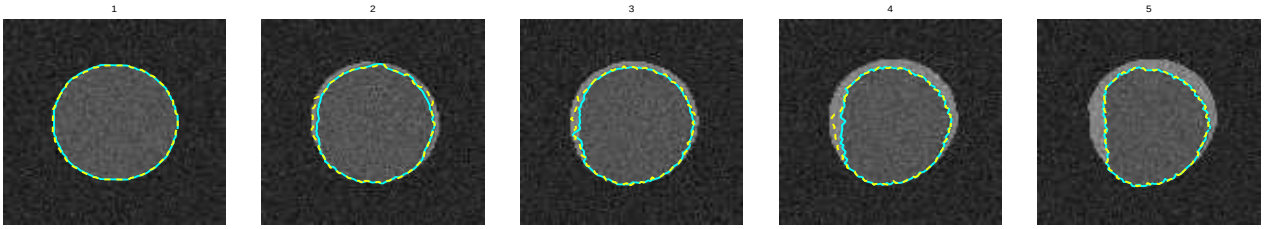
Learn A_ρ, Σ_ρ using $\{\rho_n\}$: Learn exactly as above.

Learn Δ using $\{v_{n,r}\}$: Estimate $\Delta = \frac{1}{(T-1)} \sum_{n=2}^{T} \frac{1}{M_{n-1}} v^T_{n,r} v_{n,r}$. This uses Assumption 4.

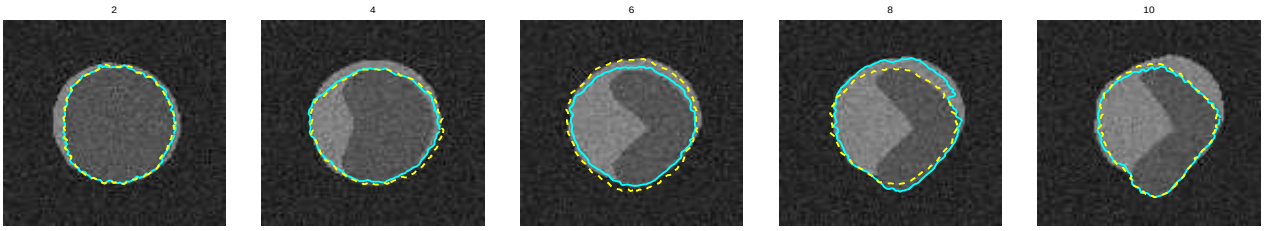
V. Simulation and Experimental Results

Since the posterior can be multimodal, plotting the "average" contour is not useful. In all the figures, we plot

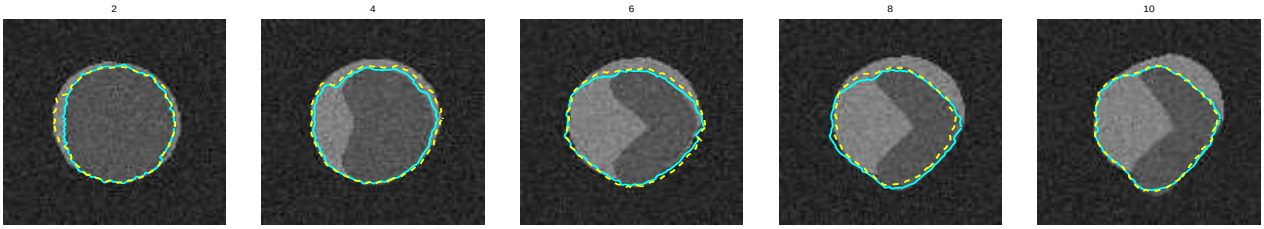
[5]Correlation depends on the distance between the basis points and hence may be different for training and test data. It is thus better to set it to zero.



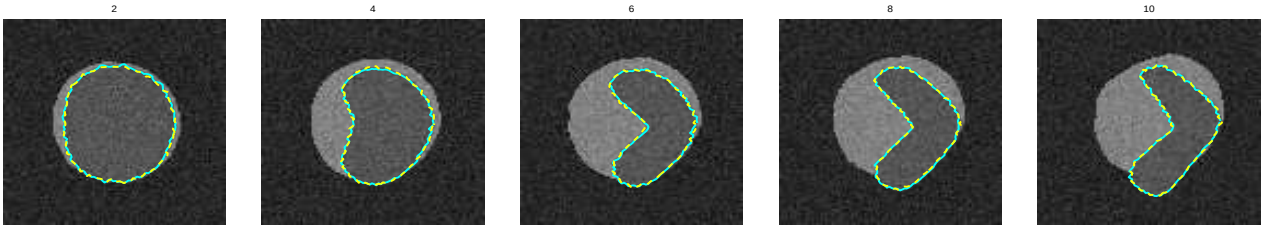
(a) Small non-affine deformation. Affine PF-MT [27] with $k = 6$ Gradient Descent (GD) iterations in Mode Tracking step: works



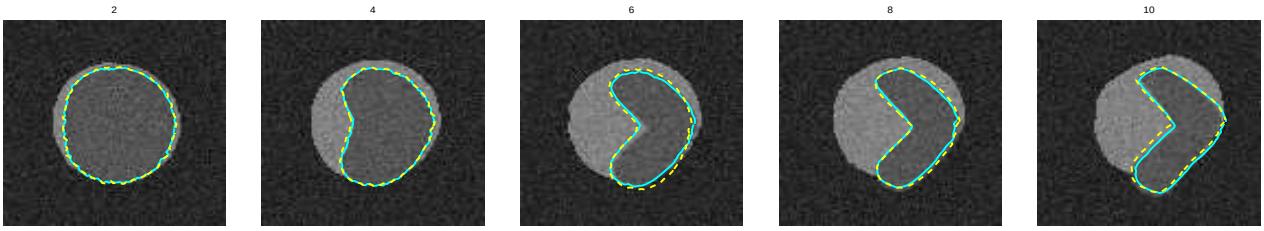
(b) Large non-affine deformation. Affine PF-MT [27] with $k = 6$ GD iterations in Mode Tracking step: fails



(c) Large non-affine deformation. Affine PF-MT [27] with $k = 12$ GD iterations: fails



(d) Large non-affine deformation. Algorithm 2 with $k = 6$ GD iterations in Mode Tracking step: works

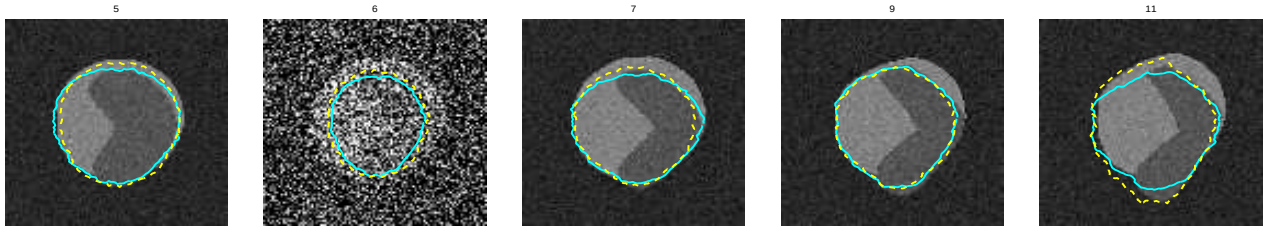


(e) Large non-affine deformation. Algorithm 2 with $k = 2$ GD iterations: works

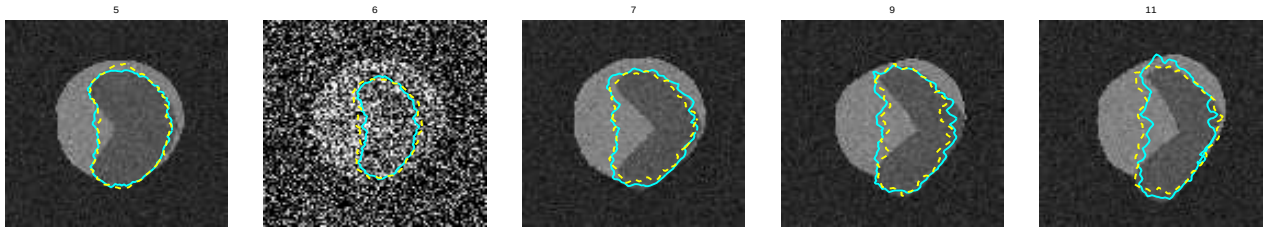
Fig. 3. Tracking through two non-affine modes. Goal is to track the dark grey object. 3(a): Non-affine deformation per frame of the dark grey object is small and hence [27] is able to track it. 3(b),3(c),3(d),3(e): Non-affine deformation per frame is large and hence [27] fails even with increasing the GD iterations. But Algorithm 2 works with as low as 2 GD iterations. $K = 6$ knots (basis points) were used.

two contours with the largest posterior (largest weights). The largest weight contour is shown as a solid cyan line, the second one as a dotted yellow line. In Figs. 3 and 4, we demonstrate simulated examples of situations where the algorithm of [27] (we call it Affine PF-MT) will fail but Algorithm 2 will work. In Fig. 5 we demonstrate the need to change K as contour length changes and track with PF-MT-TV (Algorithm 3). This uses the parametric basis, while all other experiments use the geometric basis. We show results for tracking two contour modes of a car due to partial occlusion by a light pole in Fig. 6. Tracking of a brain tumor boundary and of the right ventricle through sequential MR slices is shown in Fig. 7.

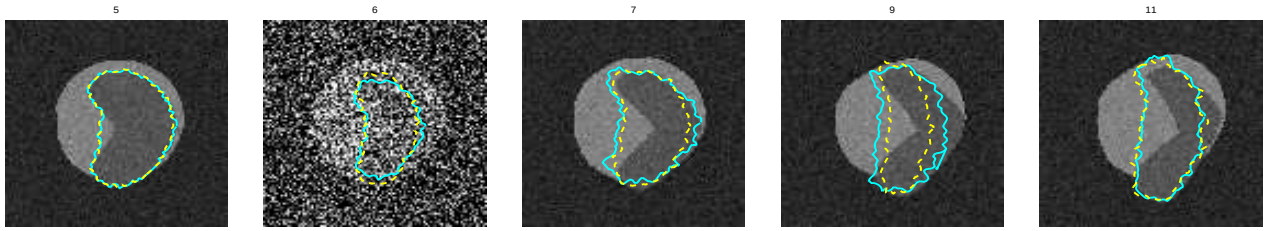
In Fig. 3(a), we show a sequence with two contour modes that are separated by non-affine deformation (due to overlapping objects). But the non-affine deformation per frame is small and so is correctly tracked using Affine PF-MT [27]. In Figs. 3(b)-3(e), the object of interest (dark grey) has large non-affine deformation per frame and so [27] fails while Algorithm 2 works. We simulated the image sequences as follows. The contour of the background (light grey) object was simulated by starting with a circle at $n = 0$ and using the system model described in (7)-(10) with $K = 6$ (corresponds to $\alpha_s = 2\pi/6$) to move and deform it. A geometric basis was used here. We used $\Sigma_s = I_K$, $A_s = 0.5I_K$ and $\Sigma_\rho = 0.25I_2$. The



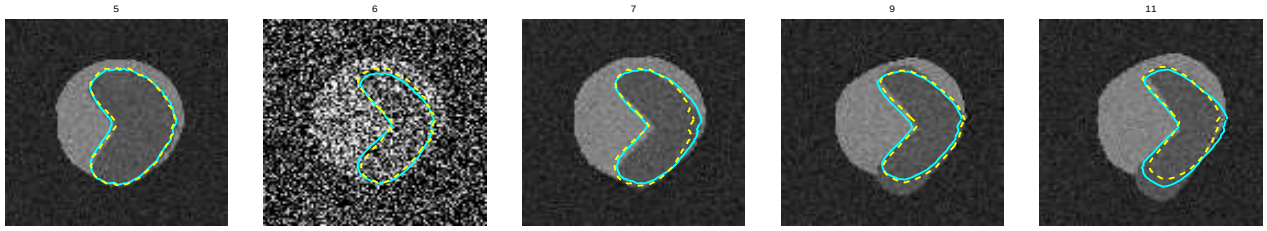
(a) Affine PF-MT [27] with $k = 2$ GD iterations in Mode Tracking step: fails



(b) Affine PF-MT [27] with $k = 6$ GD iterations and ensuring that frame 5 is in track (done by removing GD towards edge energy): fails

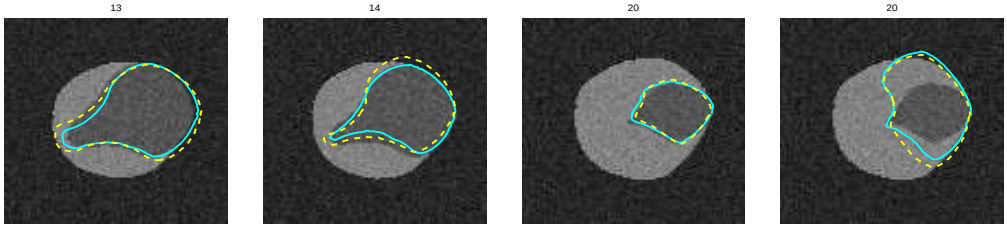


(c) Affine PF-MT [27] with $k = 12$ GD iterations and ensuring that frame 5 is in track: fails



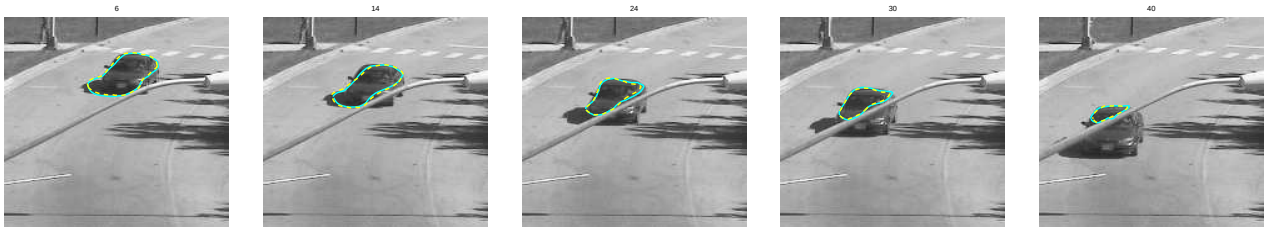
(d) Algorithm 2 with $k = 2$ GD iterations in Mode Tracking step: works

Fig. 4. Tracking through outlier observations. Starting at $n = 6$, every even frame was an outlier similar to frame 6 shown in the second column. This was done by added observation noise with $\sigma^2_{obs} = 10000$. 4(a): [27] loses track even before outliers occur. 4(b),4(c): we ensure that frame 5 remains in track (done by removing GD towards edge energy) and use more GD iterations, but [27] still loses track. Increasing GD iterations seems to only worsen the loss of track (contour attracted more towards outlier), e.g. in 4(c) (last two columns) the right boundary is more out of track than in 4(b). 4(d): Algorithm 2 remains in track until $n = 9$, very slight loss of track after that. This demonstrates the fact that one needs to importance sample on the space of local deformations to ensure no loss of track.

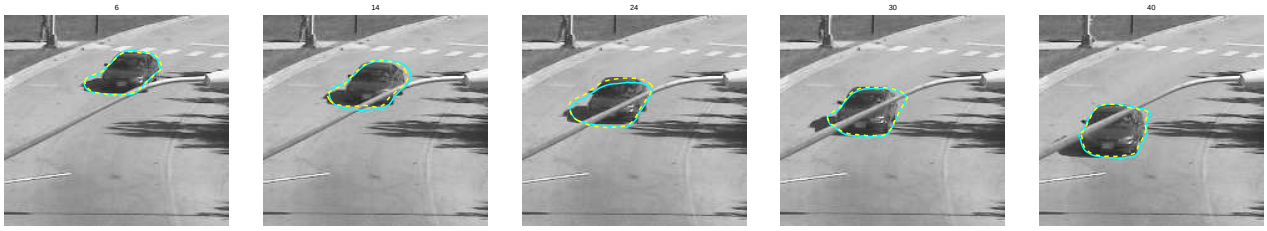


Frame 13, $K = 6$ Frame 14, $K = 5$ Frame 20, $K = 4$ Frame 20, $K = 6$

Fig. 5. The need to change K. The dark grey object deforms and keeps reducing in size which requires reducing K. The first three columns are tracked using Algorithm 3. The tracking is not perfect because only N=15 particles were used. In the last column, we show what happens if we keep track using Algorithm 2 with $K = 6$ fixed. Some contours develop self intersections resulting in zero weight assigned to them (not shown). The ones with non-zero weight are those which did not self-intersect because they started expanding instead (shown).

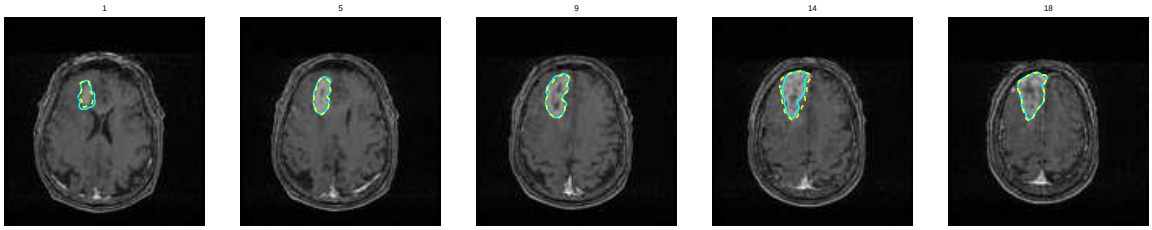


(a) Tracking the contour for the car to left of the pole using Algorithm 2

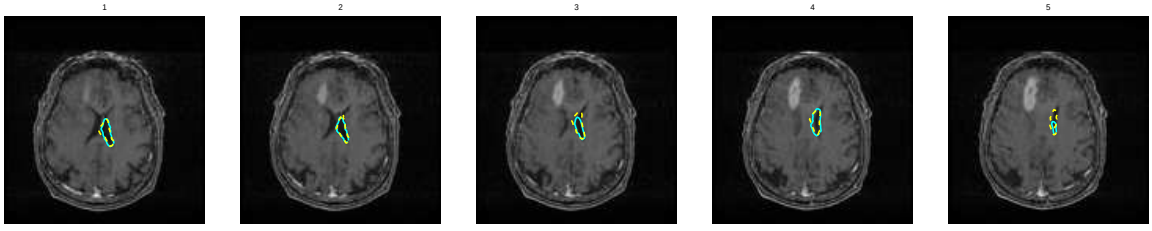


(b) Tracking the full car using Algorithm 2

Fig. 6. Tracking a car through partial occlusion by a pole. For 6(a) (left part of car), v_1=average intensity of pole, v_2=average intensity of road and u_1=average intensity of the car. For 6(b) (full car), $v_1 = v_2$ =average intensity of road but u_1=average intensity of the car and u_2=average intensity of pole. The tracking of the full car is not as accurate because we do not enforce closeness to a rigid car's template as is done in [27] and [17].



(a) Tracking the tumor (grey-white region) through sequential MR slices of brain using Algorithm 2. Notice the low contrast imagery.



(b) Attempt to track the right ventricle (black region in the center) using Algorithm 2. Notice the low contrast imagery.

Fig. 7. Tracking the tumor (grey-white region) and the ventricle (black region in the center) in a brain MRI sequence. The tumor tracking is accurate even without having learnt the system model. The ventricle tracking can improve a lot with learning the prior model of its shape change over slices.

residual deformation was set to zero while simulating, i.e. we set $C_n = \tilde{C}_n$. The contour of the dark grey deforming object (object of interest) was also simulated with a model similar to the one above with the difference that a non-zero drift term, μ, was added in equation (8). We used $\mu = [0\ 0\ 0\ 0\ 0\ 1]^T$ in Fig. 3(a) (small deformation per frame) but we used $\mu = [0\ 0\ 0\ 0\ 0\ 2]^T$ in Fig. 3(b)-3(e) (large deformation per frame). This introduced a non-zero bias in the velocity dynamics of the 6^{th} knot (basis point), resulting in an inward motion with non-zero average velocity at any n. The intensity of each pixel inside the contour of the object of interest (dark grey) was taken to be i.i.d Gaussian distributed with mean $u_1 = 85$ and variance $\sigma^2_{obs,r} = 100$. The mean intensity of the background (light grey) object was $v_1 = 130$ and variance was $\sigma^2_{obs,r} = 100$. The outer (black) background had mean intensity, $v_2 = 45$ and variance, $\sigma^2_{obs,r} = 100$.

We compared the tracking of the dark grey object using Algorithm 2 (Fig. 3(d), 3(e)) with that using Affine PF-MT [27] (Fig. 3(b), 3(c)). We used $N = 45$ particles in both cases. Observation likelihood was defined as explained in Section II-D. The edge term was taken from [6]. It had two equally strong modes at the background and foreground objects. The region term was the Chan-Vese model [45], [27] with one difference: the background could have two possible intensities v_1 and v_2. This is explained in the Appendix. It had a strong mode only at the object of interest. Thus the combined likelihood had a strong mode at the object of interest and a weaker mode at the background object. When tracking with Algorithm 2, we used all simulation parameters with the exception that we set $\mu = 0$. Instead, to track the non-zero bias in the velocity, we increased the system noise variance of the 6^{th} knot to 5, i.e. $\Sigma_s = diag([1\ 1\ 1\ 1\ 1\ 5])$ was used for

tracking. Geometric basis was used. Residual deformation was tracked as explained in the Mode Tracking step (step 2) of Algorithm 2 with $k = 6$ Gradient Descent (GD) iterations in Fig. 3(d) and $k = 2$ GD iterations in Fig. 3(e). As can be seen, even 2 GD iterations suffice. In Fig. 3(b), 3(c), we show tracking of the same sequence using Affine PF-MT [27]. This algorithm used the space of affine deformations as the effective basis and all non-affine deformation was treated as "residual deformation" (tracked using a method similar to Mode Tracking of Algorithm 2). We show results with $k = 6$ and $k = 12$ GD iterations. Since there are two distinct modes with roughly the same affine deformation (w.r.t. a circle) and non-affine deformation per frame is large, the contours get stuck to the wrong mode in the Mode Tracking step. Increasing GD iterations does not help. We would like to clarify that we did not learn the affine deformation parameters (and hence it is not a fair comparison), but we changed the values a number of times until best possible results were obtained.

The sequence of Fig. 4 was also generated exactly as above, but with outlier observations. Outlier observations similar to the one shown in the second column were simulated at every even frame starting at $n = 6$. This was done by increasing the observation noise to $\sigma^2_{obs,r} = 10000$ in the even frames. Before $n = 6$, the dark grey object is well approximated by affine deformation of a circle and hence is in track using both algorithms (if enough GD iterations are used for Affine PF-MT[27]). But at and after $n = 6$, [27] gets stuck in the wrong mode due to the outlier observation. Since it does not generate samples for local deformation, it is unable to get back to the correct mode before the next outlier appears. For this example, increasing GD iterations only worsens the loss of track. On the other hand, the tracking with Algorithm 2 is shown in Fig. 4(d). It is able to get back to the correct mode quickly, because it samples the space of non-affine deformations. It uses only $k = 2$ GD iterations.

In Fig. 5, we demonstrate the need to change the effective basis dimension, K. In the image sequence shown, the contour length keeps reducing because of inward motion of knots 3 and 5 (simulated by using a non-zero drift $\mu = [0\ 0\ 1\ 0\ 1\ 0]$). We used a parametric basis here and $\alpha_s = 35$. While generating the sequence, K reduces from 6 to 4 at $n = 15$ and to 3 at $n = 23$. While tracking using Algorithm 3, we detected the need to reduce K from 6 to 5 at $n = 14$, from 5 to 4 at $n = 17$ and to 3 at $n = 23$. The results are shown in the first three columns. In the last column, we show what happens if we do not allow change in K, i.e. we track with Algorithm 2 with $K = 6$ fixed. When the knots come too close, independent velocity samples at these points often erroneously result in a contour with self-intersections (which breaks). All such contours get assigned zero weights. The contour particles that remain with non-zero weights are those which started expanding erroneously.

Fig. 6 shows a moving car going under a street pole which partially occludes it for some frames. One may want to track the full car or track the portion to the left of the pole or the right portion of the car. We demonstrate the first two cases. The left part of the car was tracked by using v_1=average intensity of pole, v_2=average intensity of road and u_1=average intensity of the car in the region-based term of the observation likelihood. Tracking results are shown in Fig. 6(a). For tracking the full car, we used, v_1=v_2=average intensity of road but u_1=average intensity of the car and u_2=average intensity of pole. This is shown in Fig. 6(b). The tracking of the full car is not as accurate because we do not enforce closeness to a rigid car template as is done in [27] and [17].

Fig. 7 shows sequential segmentation of a set of MRI slices of different cross-sections of the brain. We show results on segmenting brain tumor (grey-white region) in Fig. 7(a). The low contrast in the images results in a large number of weak observation likelihood modes, very near the true one. There is intensity variation across the sequence and hence the edge likelihood helps remain in track. Preliminary results on sequentially segmenting the right ventricle (inside black region) are shown in Fig. 7(b). We have not learnt the contour dynamics here and hence the poor segmentation of the ventricle. Learning its dynamics as explained in Section IV will help improve the results a lot. This is especially true for normal brain regions such as the ventricle which have similar shapes across patients.

VI. Conclusion and Open Issues

A new algorithm for tracking deforming contours is proposed, which uses the fact that in most problems, at any given time, most of the contour deformation occurs in a small number of dimensions (effective basis) while the deformation in the rest of the dimensions (residual space) in small. A modification of the standard PF, called PF-MT, is proposed which importance samples the x-y translation velocity and the deformation velocity on a 6 dimensional effective basis (followed by interpolation onto the entire contour) while replacing importance sampling by deterministic posterior mode tracking (MT) in the residual space. PF-MT-TV which can handle effective basis dimension change is also proposed and implemented. Note that our algorithms can also be used with other types of effective basis, e.g. PCA basis [29], [9], [22] and also with other representations of the contour (other than level sets). The PF-MT idea can also be applied to other problems, e.g. tracking optical flow or tracking illumination change.

There are many open issues. The appropriate choice of effective basis is not clear. Geometric basis is easy to implement and handles topology change, but there is a non-uniqueness problem if two or more points on the contour have the same radial angle. The parametric basis (velocity as a function of arclength) handles this issue, but cannot deal with topology change. A second issue is the choice of effective basis dimension, K, and how to change K for both types of bases. Some initial ideas are introduced in Section IV. When changing K while tracking, one also needs to deal with errors in estimating K, for e.g. using the ideas introduced in [28].

Another open issue for any PF problem with multimodal posteriors is which posterior state estimate to display (mean or largest mode or all modes). Computing all modes on large dimensions is usually difficult. One heuristic is to assume that different contour modes will either be separated by translation or if they are separated by deformation, then they will have different contour lengths. A very important implementation issue is the choice of observation models and the use of efficient resampling techniques [51], [52],[4, Ch. 12,13], for

large dimensional problems. For e.g., the region based observation model can be improved by also tracking object and background intensities as in [9], to allow for illumination variations over time. Application to medical image sequence segmentation problems, e.g. tracking different regions of an organ such as the brain or the heart, from an MRI sequence or from a more noisier ultrasound sequence, is currently being explored. For most medical image sequences, large amounts of hand-segmented training data can be obtained. Current experimental results are without using the learning algorithm of Section IV, but learning can greatly improve the results.

APPENDIX

A. Geometric and Parametric Effective Basis

1) Geometric Basis: We use the radial angle (angular coordinate of the contour point w.r.t. the centroid of the contour's inside region, $[\mu_n^y, \mu_n^y]$), as the parameter. Thus

$$B_s(C_n)(p) \triangleq B_s(\underline{\theta}^*)(\theta(C_n(p))), \text{ where}$$
$$\theta(C_n(p)) \triangleq \arctan[\frac{C_n^x(p) - \mu_n^x}{C_n^y(p) - \mu_n^y}] \tag{16}$$

and the vector $\underline{\theta}^*$ contains K basis points (called "knots" [53]) which are uniformly chosen at angular distance $\alpha_s = 2\pi/K$ apart. $B_s(\underline{\theta}^*)(p)$ contains the K B-spline basis functions for a closed cubic B-spline [53] with knots $\underline{\theta}^*$.

2) Parametric Basis: This parameterizes velocity based on the arclength [5] of the contour point, $s(C_n(p))$, w.r.t. an initial starting point. We initially place the knots on the contour uniformly at $\alpha_s = L/K$ arclength distance apart where L is the contour length. As the contour deforms, the knots also move on the contour. Thus for the parametric basis,

$$B_s(C_n)(p) \triangleq B_s(\underline{s}(\underline{x}_n^*, C_n))(s(C_n(p))) \tag{17}$$

where $\underline{x}_n^*$ has components, $\underline{x}_{n,j}^*, j = 1, 2, ..K$ which denote the x-y location of the j^{th} knot. The vector $\underline{s}$ has components $\underline{s}_j, j = 1, ..K$ which denote the arclength location of the j^{th} knot w.r.t. a fixed starting point. Given a contour, C, there is an invertible mapping between $\underline{x}^*$ and $\underline{s}$. The forward mapping, $\underline{s}(\underline{x}^*)$ is: $\underline{s}_1(\underline{x}^*, C) = 0$ and

$$\underline{s}_j(\underline{x}^*, C) = \underline{s}_{j-1} + \frac{1}{L} arclen(\underline{x}_j^*, \underline{x}_{j-1}^*, C), \text{ for } j = 2, 3, ..K$$

$$arclen(\underline{x}_j^*, \underline{x}_{j-1}^*, C) \triangleq \sum_{m=m_{j-1}}^{m_j} ||C(p_m) - C(p_{m-1})||, \text{ where}$$
$$m_j \triangleq \arg\min_m ||\underline{x}_j^* - C(p_m)|| \tag{18}$$

Here $arclen()$ is the arclength [5] between the two consecutive knot locations, $\underline{x}_j^*, \underline{x}_{j-1}^*$. The inverse mapping is:

$$\underline{x}_j^*(\underline{s}, C) = C(\underline{s}_j). \tag{19}$$

The knot locations, $\underline{x}_n^*$ move along with the contour, i.e. for all $j = 1, 2, ..K$, they follow:

$$\underline{x}_{n,j}^* = \underline{x}_{n-1,j}^* + v^*(p_j)\overrightarrow{\mathbf{N}}(p_j), \; p_j \triangleq \underline{s}_j(\underline{x}_{n-1}^*, C_{n-1}) \tag{20}$$

$v^*(p) = B_s(p)v_{n,s} + \overrightarrow{\mathbf{N}}_{n-1}^T(p)\rho_n$ is the term inside [ ] on the right hand side of (7). After sometime, some of the knots may come "too close" to each other, while others may go "too far" away. This requires a change in effective basis (Section III-D).

B. Explaining the Observation Model

The observation at time n, $Y_n = [Y_n^r, Y_n^e]$ where Y_n^r denotes the image and Y_n^e denotes the edge map (locations of all edges detected using Canny's, method). The region based likelihood is a slight modification of the Chan-Vese model [45], [27] and the edge term is similar to the model used in Condensation [6]. The observation model can be expressed as follows:

$$p(Y_n|C_n) = p(Y_n^e|C_n)p(Y_n^r|C_n) \quad \text{where}$$
$$p(Y_n^e|C_n) \propto \exp[-E_{ed}(Y_n^e, C_n)], \quad \text{where}$$
$$E_{ed} \triangleq \frac{1}{\sigma_{obs,e}^2}[\sum_{j=1}^{M_n} \min[\rho, \min_{l=1,2,..P_n} ||Y_n^e(l) - C_n(p_j)||^2] + (P_n - M_n)^+\rho], \tag{21}$$
$$p(Y_n^r|C_n) \propto \exp[-E_{cv}(Y_n^r, C_n)], \quad \text{where}$$
$$E_{cv} \triangleq \frac{1}{\sigma_{obs,r}^2} \sum_{x_1=1}^{S_1} \sum_{x_2=1}^{S_2} [\chi_{C_n,in}(x)(Y_n^r(x) - u_1)^2 + (1 - \chi_{C_n,in}(x)) \min_{l=1,2}(Y_n^r(x) - v_l)^2] \tag{22}$$

where $S_1 \times S_2$ is the image size, P_n is the number of edges in the edge map and $\chi_{C_n,in}(x)$ is the indicator function for the region inside the contour C_n. Other parameters are explained while we list the implicit assumptions of the above model.

1) Combined likelihood: The image, Y_n^r, and the edge map, Y_n^e, are independent given C_n. This assumption is obviously not true but is a convenient way to combine both likelihoods.

2) Assumptions for the edge term (21): are as follows.

a) Edge points are generated either by the object contour or by clutter. Each contour point may either generate no edge(missed detection), or may generate any one of the edge points [6]. More than one contour point may generate the same edge.

b) The location of an edge point generated by a contour point is i.i.d. Gaussian distributed about the contour point with variance $\sigma_{obs,e}^2$.

c) The edge points generated by clutter can occur anywhere in the image with uniform probability, $e^{-\frac{\rho}{\sigma_{obs,e}^2}}$. Since the image size is $S_1 \times S_2$ pixels, we get $\rho = \sigma_{obs,e}^2 \log_e(S_1 S_2)$.

d) To compute the exact edge likelihood, one would need to evaluate $(P_n + 1)^{M_n}$ likelihood terms corresponding to the different permutations linking a contour point to an edge point and sum them. This will be very expensive. The above model assumes that the observation likelihood given the most likely permutation is much larger than the rest. Thus it implicitly assumes that (i) the distance of the edge point closest to a contour point is much smaller than that of any other edge point and (ii) when a contour point is missed (not detected), its distance from all edge points is much larger than ρ.

3) Assumptions for the region term (22): are as follows.

a) The image is separated into object and background region by the contour. The pixels in the object region are i.i.d. Gaussian distributed with mean intensity u_1 and variance $\sigma_{obs,r}^2$.

b) The background pixels are also i.i.d. Gaussian with variance $\sigma_{obs,r}^2$, but have one of two possible mean values, v_1, v_2.

c) v_1 and v_2 are assumed sufficiently separated compared to $\sigma_{obs,r}^2$, so that the likelihood given the most likely permutation is much larger than the rest.

4) Spatially dependent observation noise: The spatially independent observation noise assumption is never true in practice. For the region term, we instead assume in implementation, that $a \times a$ regions (e.g. $a = 4$) in the image have the same observation noise. For the edge term, we replace the summation over all M_n contour points by summation over a subsampled set of M_n/a points.

REFERENCES

[1] N. Vaswani, A. Yezzi, Y. Rathi, and A. Tannenbaum, "Time-varying finite dimensional basis for tracking contour deformations from image sequences," in *IEEE Conf. Decision and Control (CDC)*, 2006.

[2] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE Proceedings-F (Radar and Signal Processing)*, pp. 140(2):107–113, 1993.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] A. Doucet, N. deFreitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[5] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*, Cambridge University Press, January 2001.

[6] M. Isard and A. Blake, "Condensation: Conditional Density Propagation for Visual Tracking," *Intl. Journal of Comp. Vision*, pp. 5–28, 1998.

[7] A. Blake and M. Isard, Eds., *Active Contours*, Springer, 1998.

[8] B. Georgescu, X. S. Zhou, D. Comaniciu, and B. Rao, "Real-time multi-model tracking of myocardium in echocardiography using robust information fusion," in *Intl. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2004.

[9] W. Sun, M. Cetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, and A. Willsky, "Segmenting and tracking the left ventricle by learning the dynamics in cardiac images," *MIT Technical Report 2642*, Feb 2005.

[10] S. K. Warfield, "Real-Time Image Segmentation for Image-Guided Surgery," in *Surgical Planning Lab (SPL, Harvard) webpage: http://www.spl.harvard.edu:8000/pages/papers/warfield/sc98/index.html*.

[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.

[12] D. Terzopoulos and R. Szeliski, *Active Vision*, chapter Tracking with Kalman Snakes, pp. 3–20, MIT Press, 1992.

[13] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[14] A.Blake, M.Isard, and D.Reynard, "Learning to track curves in motion," in *IEEE Conf. Decision and Control (CDC)*, 1994.

[15] R.W. Brockett and A. Blake, "Estimating the shape of a moving contour," in *IEEE Conf. Decision and Control (CDC)*, 1994.

[16] J. Jackson, A. Yezzi, and S. Soatto, "Tracking deformable moving objects under severe occlusions," in *IEEE Conf. Decision and Control (CDC)*, 2004.

[17] M. Niethammer and A. Tannenbaum, "Dynamic level sets for visual tracking," in *IEEE Conf. Decision and Control (CDC)*, 2004.

[18] A. Mansouri, "Region tracking via level set pdes without motion computation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no.7, pp. 947–961, July 2002.

[19] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 11, 2004.

[20] T. Zhang and D. Freedman, "Tracking objects using density matching and shape priors," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 2003, pp. 1950–1954.

[21] Y. Shi and W.C. Karl, "Real-time tracking using level sets," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[22] D. Cremers, "Dynamical statistical shape priors for level set based tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, August 2006.

[23] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[24] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 2nd edition, 1999.

[25] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer Verlag, 2003.

[26] N.Vaswani, A.Yezzi, Y.Rathi, and A.Tannenbaum, "Particle filters for infinite (or large) dimensional state spaces-part 1," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[27] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Particle filtering for geometric active contours and application to tracking deforming objects," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[28] N.Vaswani, "Particle filters for infinite (or large) dimensional state spaces-part 2," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[29] M. Leventon, E. Grimson, and Olivier Faugeras, "Statistical shape influence in geodesic active contours," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 1316–1324.

[30] A. Tsai, A. Yezzi, W. Wells III, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky, "Model-based curve evolution technique for image segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001, vol. 1, pp. 463–468.

[31] O.Juan, R.Keriven, and G.Postelnicu, "Stochastic mean curvature motion in computer vision: Stochastic active contours," in *VLSM*, 2004.

[32] K.Toyama and A.Blake, "Probabilistic tracking in a metric space," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 2001, pp. 50–59.

[33] M. Bertalmo, G. Sapiro, and G. Randall, "Morphing active contours," *PAMI*, vol. 22, no. 7, pp. 733–737, July 2000.

[34] M. Bertalmo, G. Sapiro, and G. Randall, "Region tracking on level-sets methods," *IEEE Trans. Med. Imaging*, vol. 18(5), pp. 448–451, 1999.

[35] N. Paragois and R. Deriche, "Geodesic active contorus and level sets for the detection and tracking of moving objects," *Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.

[36] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for motion estimation and tracking," *Computer Vision and Image Understanding*, vol. 97, pp. 259–292, 2005.

[37] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. CVPR*, 2000, vol. 2, pp. 142–149.

[38] R. Chen and J.S. Liu, "Mixture kalman filters," *Journal of the Royal Statistical Society*, vol. 62(3), pp. 493–508, 2000.

[39] T. Schn, F. Gustafsson, and P.J. Nordlund, "Marginalized particle filters for nonlinear state-space models," *IEEE Trans. Sig. Proc.*, 2005.

[40] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Trans. Signal Processing*, pp. 613–624, 2001.

[41] W. Fong and S. Godsill, "Sequential monte carlo simulation of dynamical models with slowly varying parameters: Application to audio," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2002.

[42] A.Yezzi and A.Mennucci, "Conformal metrics and true "gradient flows" for curves.," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 2005.

[43] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, Inc., 1991.

[44] J. Trainsel and A. Tannenbaum, "Edge based stochastic active contours for medical imaging," in *Proc. of SPIE*, January 2006.

[45] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[46] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering," in *Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering*, 1998.

[47] G. Casella and R. Berger, *Statistical Inference*, Duxbury Thomson Learning, second edition, 2002.

[48] A. Yezzi and S. Soatto, "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *Intl. Journal of Comp. Vision*, vol. 53, no. 2, pp. 153–167, 2003.

[49] F. LeGland and N. Oudjane, "Stability and Uniform Approximation of Nonlinear Filters using the Hilbert Metric, and Application to Particle Filters," *Technical report, RR-4215, INRIA*, 2002.

[50] A.V. Oppenheim, A.S. Willsky, and I.T. Young, *Signals and Systems*, Prentice Hall of India, 1990.

[51] J.P. MacCormick and A. Blake, "A probabilistic contour discriminant for object localisation," *IEEE Intl. Conf. on Computer Vision (ICCV)*, Mumbai, India, June 1998.

[52] J.P. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 1999.

[53] David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, WCB/McGraw-Hill, 1990.